Closed-Form Pachelbel: A Deterministic, Symbolic Encoding and High-Fidelity Rendering of the Canon in D

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 31, 2025

Pachelbel's Canon in D is beloved for its eight-bar ground bass and imitative counterpoint, yet most digital realizations treat it as a sequence of notes rather than a compact mathematical object. This paper develops a closed-form, fully deterministic encoding of the Canon that maps time to pitch classes, registers, rhythms, dynamics, and microtiming—then renders it directly to performance-grade MIDI. The model starts from the period-T ground (I–V–vi–iii–IV–I–IV–V in D major), defines a discrete grid (L=16 sub-beats per bar), and specifies a delay Δ=2 bars that generates three canonic upper voices from a single contour function q(t). Harmony is enforced via chord projectors P_b, while register selection uses a nearest-neighbor operator N within voice-specific ranges. Musical phrasing is encoded by bar-arch envelopes, cycle-scale swells, and a light, deterministic micro-jitter ϕ_c(t) that yields humanlike timing without randomness. Baroque idioms (4–3 suspensions, 6–5 over V, leading-tone resolution, continuo realization) are expressed as simple tabled transforms on the same grid, preserving analytic clarity. The result is a succinct specification that reproduces the Canon's recognizable flow while remaining editable, extensible, and portable across synthesizers and sample libraries. We close by discussing evaluation, variants (pure strings, piano reduction), and how the method generalizes to other ostinato-canon forms.

Keywords: Pachelbel Canon, closed-form music encoding, symbolic music, canonic imitation, figured bass, Baroque counterpoint, MIDI rendering, microtiming, performance modeling, algorithmic composition.

A collaboration with GPT-5. 45 pages. CC4.0.

Outline

1. Motivation and Prior Art
   1.1 Why closed-form, not note lists
   1.2 Relation to symbolic/algorithmic music literature
2. Canon Structure in One Period
   2.1 Ground: (I, V, vi, iii, IV, I, IV, V) in D major
   2.2 Discrete timeline: bars, beats, L=16 grid, period T=8 bars
   2.3 Canon delay Δ=2 bars and voice indexing
3. The Formal Encoding
   3.1 Chord projectors P_b and ordered triads A_b
   3.2 Contour word q_n and rhythm set R_n per cycle
   3.3 Pitch-class selection and register operator N
   3.4 Deterministic micro-timing ϕ_c(t) and velocity envelope v(t)
4. Baroque Idioms as Local Transforms
   4.1 4–3 suspensions on I and leading-tone emphasis on V→I
   4.2 6–5 over V and continuo realizations in eighths
   4.3 Rules to reduce parallel octaves/fifths (inner-voice root avoidance)
5. Rendering Pipeline (Library-Free SMF)
   5.1 Event construction and delta-time packing
   5.2 Program/CC selection, legato overlaps, phrase breaths and ritardandi
   5.3 Determinism and reproducibility
6. Evaluation and Listening Tests
   6.1 Objective checks: voice-leading constraints, register bounds
   6.2 Subjective criteria: phrasing, clarity, cadence perception
   6.3 Variants: pure strings, piano reduction, shortened concert cut
7. Generalization
   7.1 Other ground-bass canons and passacaglias
   7.2 Importing alternative contour libraries R_n, q_n
   7.3 From Pachelbel to parametric “style surfaces”
8. Limitations and Future Work
   8.1 Ornamentation density vs. clarity
   8.2 Figured-bass grammar learning from corpora
   8.3 Orchestration beyond strings/continuo

9. Conclusion

References

Appendix A: Minimal closed-form spec (all functions and constants)
Appendix B: Pseudocode for SMF writer and event scheduling
Appendix C: Tables for idioms (suspensions, cadences, continuo patterns)

1. Motivation and Prior Art

1.1 Why closed-form, not note lists

Most digital engravings and MIDI files represent Pachelbel's Canon as a long, flat list of timestamped pitches. That is faithful but opaque: it hides the generative structure that actually *makes* the Canon the Canon. A closed-form encoding instead names the invariants and renders the rest by rule:

- **Periodicity and equivariance.** The piece is governed by an 8-bar ground with a 2-bar canonic delay. Writing it as functions over a bar-grid (period T, grid L, delay Delta) captures the translation symmetry explicitly: q(t), q(t−Delta), q(t−2Delta). Lists merely *repeat*; closed-form *explains* repetition.
- **Separation of concerns.** Harmony, rhythm, contour, register, dynamics, microtiming are each small, composable maps:
    - P_b: selects the triad for bar b in the I–V–vi–iii–IV–I–IV–V cycle.
    - R_n: a set of grid onsets per cycle n controlling density.
    - q_n: a triad-index word mod 3, choosing (root, third, fifth) locally.
    - N: a nearest-neighbor register operator with per-voice ranges.
    - v(t), phi_c(t): phrase and timing envelopes.
      This modularization lets us vary one layer (e.g., more suspensions) without disturbing others.
- **Generalization and compression.** A tiny table (chord projectors + two short rhythm/contour libraries) reproduces minutes of music at score level. That is both a conceptual compression and a practical one: the same engine can render variants (pure strings, piano reduction) by swapping only the output layer.
- **Reproducibility with interpretive nuance.** Performance realism is usually treated as post-hoc humanization. We encode phrasing and "human feel" as *deterministic* functions (bar arches, cycle swells, light micro-jitter), so identical inputs yield identical, nuanced outputs. Lists cannot separate content from interpretation so cleanly.
- **Analysis, verification, and editability.** Constraints like "avoid inner-voice roots on strong beats" or "emphasize leading tone before I" become one-line transforms applied across the whole piece. They are easy to inspect, prove, and tweak. With a note list, such global edits are brittle.

- **Faithfulness to Baroque practice.** Figured-bass habits (4–3 over I, 6–5 over V, passing-tone placement) are procedural. Closed-form is a better substrate for such procedures than static event collections.

In short: note lists *store* the Canon; closed-form *names its laws* and renders the instances.

1.2 Relation to symbolic / algorithmic music literature

Our approach sits at the intersection of several traditions; it borrows their strengths but keeps the representation minimal and deterministic.

- **Rule-based / grammar systems.** From early production systems and Schenkerian-inspired grammars to constraint-based counterpoint, the literature encodes style as explicit rules. We adopt this spirit but restrict to a few high-leverage maps (P_b, R_n, q_n, N, v, phi). Instead of a large rulebook, we use small tables aligned to the Canon's known invariants (periodicity, canonic delay).
- **Functional time representations.** Many symbolic systems move from event lists to functions of time for rhythm and harmony. We make that explicit on a fixed grid: t ↦ pitch-class via chord projectors and contour words, then register selection via N. This reduces ambiguity while staying musically transparent.
- **Canons and isorhythm.** Algorithmic treatments of canons often emphasize shift-equivariant voices and repeating taleae/color. Our construction is exactly a delay-equivariant mapping: voices are time-shifts of a single contour modulo harmony, with independent but synchronized phrasing layers. This keeps the canonical identity taut, not heuristic.
- **Figured-bass realization.** Prior systems either learn realizations from corpora or apply handcrafted rules. We adopt a lightweight, declarative layer: local 4–3 and 6–5 motions, continuo in 8ths, and voice-leading guards (e.g., inner-voice root avoidance). These are small, auditable transforms—not black-box inferences—so they travel well across renderers.
- **Performance modeling.** Expressive timing and dynamics are often stochastic ("humanization"). We instead use low-amplitude, *deterministic* envelopes tied to the bar and cycle clocks. This aligns with empirical findings that much perceived expressivity is phase-locked to musical structure rather than random.
- **Data-driven composition.** ML systems (e.g., transformer-based) can regenerate the Canon's surface but typically do not expose succinct controls for symmetry, delay, or figured-bass logic. Our closed-form is complementary: it can seed, constrain, or evaluate

data-driven models, and, conversely, corpora can refine the small tables (R_n, q_n) while the functional skeleton remains intact.

- **Information-theoretic views.** Several works measure musical redundancy and motif reuse. Our encoding is an explicit *compression* that factors recurrent structure (period, delay, triad orbit) from ornament. It is thus a constructive example of "explainable compression" in symbolic music.

Positioned this way, the contribution is not another grand theory, but a *practical minimal core* for this specific work: a handful of named functions that (i) recover the Canon's identity, (ii) carry Baroque idiom as local transforms, and (iii) render directly to high-fidelity MIDI—deterministically.

2. Canon Structure in One Period

2.1 Ground: (I, V, vi, iii, IV, I, IV, V) in D major

We fix pitch-class numbers: C=0, C#=1, D=2, D#=3, E=4, F=5, F#=6, G=7, G#=8, A=9, A#=10, B=11.

D-major diatonic triads (ordered as root, third, fifth):

- I (D): [2, 6, 9] = D, F#, A
- V (A): [9, 1, 4] = A, C#, E
- vi (Bm): [11, 2, 6] = B, D, F#
- iii (F#m): [6, 9, 1] = F#, A, C#
- IV (G): [7, 11, 2] = G, B, D

Ground (8 bars, indexed b=0..7):
C_seq[b] = [I, V, vi, iii, IV, I, IV, V].

Define a chord projector P_b that, for any pitch class x, returns 1 if $x \in$ C_seq[b] and 0 otherwise. Let A_b = (a_b(0), a_b(1), a_b(2)) be the ordered triad for bar b, so a_b(j) selects root/third/fifth via $j \in \{0,1,2\}$.

The bass (ground) root sequence is:
root_seq[b] = [2, 9, 11, 6, 7, 2, 7, 9] (D, A, B, F#, G, D, G, A).

2.2 Discrete timeline: bars, beats, L=16 grid, period T=8 bars

Timing constants:

- beats_per_bar = 4
- ticks_per_beat = Q (e.g., Q=480 in the render)
- bar length T_bar = 4*Q ticks
- grid subdivision L = 16 per bar (sixteenth-note grid)
- grid tick g = T_bar / L
- period T = 8*T_bar ticks (one full ground cycle)

Bar and grid indices:

- Bar index: b(t) = floor(t / T_bar) mod 8
- Intra-bar grid index: i(t) = floor((t mod T_bar) / g), so i ∈ {0,..,15}

Lead voice activation at time t is governed by two small tables per cycle n:

- Rhythm set R_n ⊆ {0,..,15}
- Contour word q_n: {0,..,15} → {0,1,2}

On grid points (i ∈ R_n), the **proposed** pitch class is:
pc_prop(t) = a_{b(t)}(q_n(i(t))).

Registering to MIDI:
Let N_k(pc, prev) choose the nearest MIDI note with pitch class pc within a fixed register window for voice k (e.g., Vln1 high, Vln2 mid, Vln3 lower), preferring small motion from prev.
The realized note is:
m_k(t_on) = N_k(pc_prop(t_on), prev_k).

Dynamics and microtiming are deterministic functions of structure:

- Velocity envelope v(t) = v0 * f_bar((t mod T_bar)/T_bar) * f_cycle(n)
- Micro-jitter (small): phi_k(t) = round(a1*sin(2π t/(7g)+c_k) + a2*sin(2π t/(11g)+c_k/2))
  Onset = t_on + phi_k(t_on). Offsets use legato/shorten rules tied to density |R_n|.

2.3 Canon delay Delta=2 bars and voice indexing

Canon delay:

- Delta = 2*T_bar ticks (two bars)

Voice definitions (delay-equivariant):

- Lead (k=1): v1(t) plays m_1(t) where i(t) ∈ R_n
- Second (k=2): v2(t) = time-shift of v1 by Delta, then re-registered:
  v2(t) := N_2((v1(t−Delta) mod 12), prev_2) with its own register window
- Third (k=3): v3(t) := N_3((v1(t−2*Delta) mod 12), prev_3)

Thus a single contour/rhythm generator yields three imitative voices via fixed delays, while N_k enforces register separation and smoothness.

Bass and continuo:

- Bass b(t): quarter-note roots from root_seq[b(t)], optionally with diatonic passing 8ths in dense cycles.
- Continuo c(t): broken-chord patterns (eighths) realized from A_{b(t)}, plus simple local transforms (e.g., 4–3 over I, 6–5 over V) applied on the L-grid.

Global guards (applied functionally at strong beats i ∈ {0,4,8,12}):

- Inner-voice root avoidance on strong beats to reduce unisons/octaves with bass.
- Leading-tone emphasis at bar 7 (C#) with resolution to D at bar 0.
- Suspensions: delay first attack in bars b ∈ {0,5} to create a 4–3 effect over I.

This period-T, delay-Delta construction makes the Canon a compact, closed-form map from discrete time to fully-voiced, expressive parts.

3. The Formal Encoding

3.1 Chord projectors P_b and ordered triads A_b

Pitch-class numbers: C=0, C#=1, D=2, ..., B=11.

Let bars be indexed by b in {0,..,7} (one period). The ground in D major:

- I(D) = {2,6,9}
- V(A) = {9,1,4}
- vi(Bm)= {11,2,6}
- iii(F#)= {6,9,1}

- IV(G) = {7,11,2}

Chord sequence per bar:
C_seq[b] = [I, V, vi, iii, IV, I, IV, V].

Define the projector P_b: {0,..,11} -> {0,1} by
P_b(x) = 1 if x in C_seq[b], else 0.

Fix an ordered triad A_b = (a_b(0), a_b(1), a_b(2)) listing (root, third, fifth). For the above, A_b is the ordered version of C_seq[b] shown explicitly in 2.1.

Bass roots:
root_seq[b] = (2,9,11,6,7,2,7,9).

3.2 Contour word q_n and rhythm set R_n per cycle

Time is integer ticks t in Z. One bar length T_bar (ticks). Grid length L=16 per bar, grid step g = T_bar / L. One period T = 8*T_bar.

Indices:

- Bar: b(t) = floor(t / T_bar) mod 8
- Intra-bar grid: i(t) = floor((t mod T_bar) / g) in {0,..,15}
- Cycle index (periods since start): n(t) = floor(t / T)

For each cycle n we specify two finite objects:

- Rhythm set R_n subset {0,..,15}. These are the grid positions that are “on” within a bar for the lead generator in cycle n. (Examples used: quarters {0,4,8,12}, eighths {0,2,4,6,8,10,12,14}, etc.)
- Contour word q_n: {0,..,15} -> {0,1,2}. On a hit i in R_n, q_n(i) chooses which element of A_b to project (0=root, 1=third, 2=fifth).

Proposed pitch class at a lead onset t_on (i.e., i(t_on) in R_{n(t_on)}):
pc_prop(t_on) = a_{ b(t_on) }(q_{ n(t_on) }(i(t_on))).

Idiomatic local transforms (optional) are small maps applied to pc_prop at specific structural sites (e.g., suspension delay on bars 0 and 5, 6–5 or 4–3 motions in continuo, leading-tone preference on bar 7). These are simple, bar-local functions tau_k that modify pc_prop before registration; see 3.4 for scheduling and 4.x for the rule table.

3.3 Pitch-class selection and register operator N

Each sounding voice k has a fixed MIDI register window [L_k, H_k] and an inertia preference toward small motion.

Let N_k(pc, prev) be the nearest MIDI note m in [L_k, H_k] such that m mod 12 = pc, minimizing |m−prev| (tie-break by higher m, or by proximity to a preferred octave center if prev is None).

Lead realization at onset t_on:
m_1(t_on) = N_1(pc_adj(t_on), prev_1),
where pc_adj(t_on) is pc_prop after any local idiom transforms tau_k and global guards (e.g., inner-voice root avoidance on strong beats when k>1).

Delay-equivariant canonic voices (k=2,3) are time shifts of the lead generator followed by independent registration:
m_k(t_on) = N_k((m_1(t_on − (k−1)*Delta) mod 12), prev_k),
*with Delta = 2*T_bar.

Bass b(t) follows root_seq[b(t)] as quarter notes (with optional passing 8ths in dense cycles), registered to a low window. Continuo c(t) arpeggiates A_{b(t)} on the L-grid (eighths), with simple 6–5, 4–3 options encoded as local tau patterns on selected grid slots.

Strong-beat guard (inner-voice root avoidance). For k in {2,3} and grid indices i in {0,4,8,12}, if pc_adj equals the bar's root a_b(0) and would collide with the bass by octave doubling, rotate to the next chord member:
pc_adj := rotate(a_b(0); A_b) where rotate maps root->third->fifth->root.

3.4 Deterministic micro-timing phi_c(t) and velocity envelope v(t)

Let c index MIDI channels/voices. All timing nuance is deterministic and phase-locked to structure.

Micro-timing (ticks):
phi_c(t) = round(A1*sin(2*pi*t/(7g) + theta_c)
*+ A2*sin(2*pi*t/(11g) + theta_c/2)),
with small amplitudes A1,A2 (e.g., <= ~2 ticks for upper strings, slightly larger for bass/continuo) and fixed channel phases theta_c. This yields "breath" without randomness and repeats coherently over the grid.

Onset/offset scheduling at a grid onset t_on:

- Scheduled onset: t_on' = t_on + phi_c(t_on).
- Nominal offset: t_off_nom = next grid onset in the same bar (or bar end).

- Articulation: choose shorten = s_dense if |R_n| is large else s_sparse, and legato overlap ol small.
- Scheduled offset: t_off' = max(t_on' + min_len , t_off_nom − shorten) + epsilon_overlap,
  with min_len a floor (e.g., 30–40 ticks), epsilon_overlap a few ticks when legato is desired.

Velocity (0..127) is a multiplicative product of a bar-shaped envelope and a slow cycle swell:

v(t_on) = clamp_0_127(V0 * f_bar((t_on mod T_bar)/T_bar) * f_cycle(n(t_on))),

where a simple, copy-safe choice is

- f_bar(x) = 0.7 + 0.3*sin(pi*x) (arch within the bar)
- f_cycle(n) = 0.9 + 0.2*sin(pi*n/(N_c−1)) for N_c cycles in the form.

Phrase breaths at period boundaries: optionally reduce tempo slightly at the last bar of each 8-bar period and resume at the next bar (small, stepwise tempo meta-events), and apply a short final ritard over the last few bars by writing a tempo schedule
{ (t_k, tempo_k) } to the meta track. Because tempo changes are structural (bar-synchronous), the micro-timing remains interpretable and reproducible.

Summary (closed-form stack):

- Harmony layer: P_b, A_b, root_seq[b]
- Generator layer: R_n, q_n
- Idiom layer: tau (suspensions, 6–5 / 4–3, leading tone on bar 7), plus strong-beat guard
- Registration: N_k with fixed [L_k,H_k] and inertia to prev_k
- Expression: phi_c(t), v(t_on), articulation rules
- Delays: m_k(t) = shift_by((k−1)*Delta) of the generator, then re-register

All maps are finite tables or elementary functions; together they define a deterministic function from integer time t to fully specified note events for each voice.

4. Baroque Idioms as Local Transforms

We encode idioms as *local maps* applied on the L=16 grid within each bar. Each transform is a small function

tau(bar b, grid i, role k, context) : pc -> pc'

that alters the proposed pitch class pc before registration N_k, without changing the global period T or the canonic delay Delta. Transforms compose left-to-right in a short pipeline (e.g., suspension -> cadential tweak -> strong-beat guard).

4.1 4–3 suspensions on I and leading-tone emphasis on V->I

Goal. Over I (D major triad), create a characteristic 4–3 suspension (G->F#) by *delaying* resolution at the bar start; approaching the final I, emphasize the leading tone C# in the previous V bar.

Sites.

- Suspension bars: b in {0,5} (the I bars within each 8-bar period).
- Cadential leading tone: b=7 (V) resolving to b=0 (I).

Rule A (suspension delay over I).

- On bar b in {0,5}, for the *upper canonic voice* k=1 only:
    - Delay the first new attack by one 8th (2 grid slots): do not sound a fresh onset at i=0; instead hold the previous note until i=2.
    - If a fresh onset must occur at i=2, prefer pc'=6 (F#) if previous held note was 7 (G), i.e., resolve 4->3 by step.
- Formal:

if b in {0,5} and k==1:

if i==0: suppress_new_onset; extend previous note to i=2

if i==2 and previous_pc==7: pc' := 6

Rule B (leading tone on V->I).

- On bar b=7 (V), for the *melodic head* (k=1) and optionally k=2:
    - At the first available onset i in R_n, force C# (pc=1); this prepares resolution to D (pc=2) at b=0.

- Formal:

if b==7 and i==min(R_n): pc' := 1

Musical effect. These two microscopic rules yield the familiar cadential “lean” (C#) and a voiced suspension that audibly resolves down (G->F#) at the start of I bars, without changing the timeline.

4.2 6–5 over V and continuo realizations in eighths

Goal. Encode light figured-bass motion in the continuo: a 6–5 suspension pattern over V and a steady arpeggiated 8th-note texture.

Continuo baseline (harpsichord).

- On each bar b, arpeggiate the ordered triad A_b as an 8th-note pattern over i in {0,2,4,6,8,10,12,14}:

pattern: [root, third, fifth, third, root, third, fifth, third]

Rule C (6–5 over V).

- When b is V and next bar is I:
    - On the *second half* of the bar (e.g., i in {8,10,12,14}), substitute the scale 6th above the bass (E, pc=4 in D major) and resolve to the scale 5th (D, pc=2).
- Minimal table, applied to those i:

if chord(b)==V and chord(b+1)==I:

  at i=8 or 12: pc' := 4 # the "6" above bass

  at i=10 or 14: pc' := 2 # resolve to the "5"

(This is a stylized local proxy for the 6–5 suspension—kept simple, bar-local, and auditable.)

Optional Rule D (4–3 color over I).

- On I bars, color early 8ths with 4->3:

if chord(b)==I:

  at i=2: pc' := 6 # 4th above bass (F# over D)

  at i=6: pc' := 4 # resolve to 3rd (E) in upper texture

Bass refinement (optional passing 8ths).

- In dense cycles (|R_n| large), split each quarter in the *bass* into two 8ths; on the back-8th move the bass by one diatonic step toward the next bar's root:

```
if dense_cycle:
  for each beat q:
    i=4q:    pc' := root_seq[b]              # on-beat root
    i=4q+2:  pc' := step_toward(root_seq[b+1])  # diatonic neighbor
```

This keeps propulsion without muddying the cadence.

4.3 Rules to reduce parallel octaves/fifths (inner-voice root avoidance)

Goal. In the inner canonic voices (k=2,3), avoid doubling the bass root on strong beats, which reduces parallel 8ves/5ths with the bass while preserving the canon.

Strong-beat guard.

- Strong grid indices: S = {0,4,8,12} (the 4 quarter beats).
- For k in {2,3}, if at (b,i in S) the proposed pitch class equals the bar's root a_b(0), rotate to the next chord member (root->third->fifth->root):

```
if k in {2,3} and i in {0,4,8,12} and pc==a_b(0):
  pc' := rotate_in(A_b, pc)  # root->third, third->fifth, fifth->root
```

Why this helps. Because the bass plays roots on the beats, inner-voice roots there tend to create vertical perfect consonances that can run in parallel with voice motion. Rotating to third or fifth preserves chord membership and the canonic contour class but decorrelates the vertical stack on strong beats.

Interaction order.
Apply idioms in a fixed, local order so they are predictable and composable:

1. cadential emphasis (Rule B),
2. suspensions (Rule A),
3. continuo 6–5 / 4–3 (Rules C/D, continuo only),
4. strong-beat guard (inner voices, 4.3),
5. registration N_k and articulation.

Copy-safe pseudo-code (per onset).

```
pc := pc_prop(b,i)

if (b==7 and i==min(R_n)): pc := 1        # leading tone

if (b in {0,5} and k==1 and i==0): hold_prev_to(i=2); skip_attack()

if (b in {0,5} and k==1 and i==2 and prev_pc==7): pc := 6


if (role==continuo and chord(b)==V and chord(b+1)==I and i in {8,12}): pc := 4

if (role==continuo and chord(b)==V and chord(b+1)==I and i in {10,14}): pc := 2

if (role==continuo and chord(b)==I and i==2): pc := 6

if (role==continuo and chord(b)==I and i==6): pc := 4


if (k in {2,3} and i in {0,4,8,12} and pc==a_b(0)): pc := rotate_in(A_b, pc)


m := N_k(pc, prev_midi)  # register & inertia

schedule_onset_offset_with(phi_k, v, articulation)
```

These transforms are intentionally *small*: they operate on a few designated grid slots and only within chord-membership. That keeps the Canon's identity intact while adding the idiomatic Baroque gestures listeners expect.

5. Rendering Pipeline (Library-Free SMF)

5.1 Event construction and delta-time packing

Goal. Turn the closed-form maps into a Standard MIDI File (SMF) without external libraries.

Clock and grid.

- Resolution: ticks_per_beat = Q (e.g., Q=480).
- One bar: T_bar = 4*Q.
- Grid: L=16, grid tick g = T_bar/L.
- Period: T = 8*T_bar.

Voice pass (per part k).

1. **Enumerate onsets.** For each bar b and grid i in R_n:
    - t_on = b*T_bar + i*g.
    - pc = a_b(q_n(i)), then apply idioms tau (Sec. 4) and guards.
    - m = N_k(pc, prev_k) // register + inertia.
2. **Microtiming.**
    - t_on' = t_on + phi_k(t_on).
3. **Duration and articulation.**
    - Let t_next be the next grid onset in the same bar (or bar end).
    - shorten = s_dense if |R_n| large else s_sparse.
    - min_len = L_min ticks (e.g., 30–40).
    - legato_overlap = ol ticks (e.g., 3–8).
    - t_off' = max(t_on' + min_len, t_next − shorten) + epsilon_overlap.
4. **Dynamics.**
    - vel = clamp_0_127(V0 * f_bar((t_on mod T_bar)/T_bar) * f_cycle(n(t_on))).
5. **Collect note events.**
    - Append (t_on', note_on(k,m,vel)) and (t_off', note_off(k,m,0)) to the part's raw list.

Sort and pack (per track).

- Sort by absolute time, then by note_off before note_on at same tick to avoid overlaps blowing polyphony, then by pitch for stability.
- Convert to **delta times**:

```
t_prev := 0
for (t_abs, msg) in sorted_events:
    dt := max(0, t_abs - t_prev)
    emit varlen(dt), msg
    t_prev := t_abs
```

- Terminate with End-of-Track: 00 FF 2F 00.

Meta track (tempo/time-sig).

- At tick 0: FF 51 03 uu vv ww (tempo, microseconds per beat), FF 58 04 nn dd cc bb (time signature).
- Phrase breaths & ritard: additional FF 51 03 events at scheduled absolute ticks (Sec. 5.2).

SMF container.

- Header “MThd”, 6-byte chunk, format 1 (multi-track), n_tracks, division=Q.
- For each track: “MTrk” + 4-byte length + packed events.

5.2 Program/CC selection, legato overlaps, phrase breaths and ritardandi

Programs (General MIDI numbers).

- Bass (cello): program 42 on ch 0.
- Harpsichord continuo: program 6 on ch 4.
- Canonic strings: program 48 (String Ensemble 1) on ch 1..3.

Emit once at track start:

00 C0 2A ; ch0 cello=42

00 C4 06 ; ch4 harpsichord=6

00 C1 30 ; ch1 strings=48

00 C2 30

00 C3 30

Expression and phrasing (CC11).

- Per-bar arch on every channel: three CC11 points per bar (start, mid, end), values e.g. [82, 98, 86].
- Emit as absolute-time CCs on each part track; they mix multiplicatively with note velocities.

Legato and articulation.

- For dense textures, shorten more (e.g., 40 ticks) to prevent smear.
- For lyrical spans, add small overlap (3–8 ticks) so consecutive notes connect.

- Always turn **note_off before next note_on** at the same pitch/time within a track to avoid stuck notes on strict players.

Phrase breaths (period cadence).

- At the last bar of each 8-bar period (bar 7), insert a small **tempo dip**:
    - bpm_slow = bpm * 0.96 at tick t=7*T_bar of each period; resume bpm at next bar.
- Write as meta tempo changes (track 0), no per-track edits required.

Final ritardando.

- Over the last 2 bars, step tempo downward in K small increments:

for k in 1..K:

t_k := t_rit_start + step*(k-1)

bpm_k := bpm * (1 - r * k/K) ; r ~ 0.12

emit (t_k, FF 51 03 tempo(bpm_k))

Pedal and reverb (optional).

- For piano reductions, add sustain pedal (CC64) curves.
- For ensemble sheen on GM synths, modest CC91 (reverb) per channel.

5.3 Determinism and reproducibility

No randomness.

- All variation derives from fixed tables and closed-form functions:
    - Harmony: A_b, root_seq[b]
    - Generator: R_n, q_n
    - Idioms: tau_k(b,i), strong-beat guard
    - Registration: N_k with fixed [L_k,H_k]
    - Expression: phi_k(t), f_bar(x), f_cycle(n)
    - Tempo plan: scheduled meta events

Stable evaluation order.

- For each onset: idioms -> guard -> N_k -> timing -> articulation -> velocity.

- For packing: sort by (t_abs, note_off_first, pitch).

Environment independence.

- No seeds, clocks, or floating nondeterminism in control flow (all sines evaluated on integer ticks; rounding rules fixed).
- Identical inputs produce **bit-identical** SMF bytes across runs and machines.

Auditability.

- Every idiom is local and table-driven; you can diff outputs by toggling transforms.
- A single **hash** of the event list (or final chunk bytes) certifies a rendering:

hash := SHA256(concatenate(track0_bytes, ..., trackN_bytes))

- Provenance tag (optional): include a text meta event with model spec/version:
    - FF 01 len "ClosedFormCanon v5: Q=480, L=16, Delta=2 bars" at tick 0.

Graceful degradation.

- If a synth ignores CC11 or tempo ramps, the piece remains correct; only phrasing nuance changes.
- If a player enforces note-on/off ordering strictly, our off-before-on sort and minimum lengths prevent dropouts.

Testing regime.

- Structural assertions on the event list:
    - No note length < min_len.
    - No negative deltas after packing.
    - Channel ranges respected (no pitch < L_k or > H_k).
    - Strong-beat guard applied when k in {2,3} and i in {0,4,8,12}.
- Round-trip: parse the emitted SMF and re-compute the above invariants.

With these constraints, the rendering is not just *musically* faithful; it is **computationally exact**, reproducible, and easy to extend—e.g., swap programs, alter idiom tables, or change grid density—without jeopardizing determinism.

6. Evaluation and Listening Tests

6.1 Objective checks: voice-leading constraints, register bounds

Static conformance (per bar b, grid i):

- **Chord membership.** For each canonic voice k in {1,2,3} and continuo/bass, verify pc_k(b,i) ∈ A_b unless a local transform tau explicitly maps to a diatonic neighbor (passing/suspension slots only).
- **Register windows.** m_k(b,i) ∈ [L_k, H_k] (fixed per voice). Flag any crossings or out-of-range notes.
- **Strong-beat guard.** On i ∈ {0,4,8,12} (quarter beats) and k ∈ {2,3}, assert pc_k ≠ a_b(0) unless harmony changes within ≤ 2 grid steps (avoid root doubling on the beat in inner voices).
- **Delay equivariance.** Check that v2 is v1 shifted by Delta=2 bars modulo register, and v3 is shifted by 2*Delta; i.e., (m_2 mod 12)(t) = (m_1 mod 12)(t−Delta), (m_3 mod 12)(t) = (m_1 mod 12)(t−2Delta).

Contrapuntal motion diagnostics (sliding windows over grid):

- **Parallel perfects.** Track interval classes between (Vln1, Vln2), (Vln2, Vln3), (Vln1, Bass). If two consecutive verticals are both 0 or 7 semitones (unison/octave) or 7/5 for perfect fifths and move in the same melodic direction, flag as parallel perfect motion.
- **Hidden perfects.** If outer voices approach perfect consonance by similar motion with a leap > 3 semitones in the upper voice, flag as potential hidden fifth/octave.
- **Dissonance placement.** When a pitch is not in A_b, assert its grid index is weak (i ∈ {2,6,10,14}) or under a labeled suspension. Resolution step direction: for 4–3 over I, check G->F#; for 6–5 over V, check E->D.

Metered content checks:

- **Bass alignment.** On each quarter onset, bass uses root_seq[b]; in “dense cycles,” the back-eighth moves one diatonic step toward next root (verify step_toward(root_seq[b+1])).
- **Continuo pattern.** Eighth-note arpeggiation matches ordered A_b except at codified transform slots (6–5 on V, optional 4–3 on I).

Timing and dynamics sanity:

- **Micro-timing bounds.** |phi_c(t)| ≤ 5 ticks (upper strings) and ≤ 6–8 ticks (bass/continuo), no sign inversions causing note overlap collapse.
- **Minimum duration.** min_len ≥ 30 ticks; after articulation shorten, still t_off' − t_on' ≥ min_len.
- **Tempo events.** Breath dips occur at bar 7 of each 8-bar period; final ritard K-step schedule is monotone.

Aggregate summaries for the appendix:

- Histogram of vertical intervals (by class) over the piece.
- Count of parallel perfect flags (expect near-zero).
- % of non-chord tones on weak vs strong grid positions (expect ≫ on weak).
- Register occupancy per voice (mean, sd, min, max).

6.2 Subjective criteria: phrasing, clarity, cadence perception

Panel and protocol.

- **Participants.** 8–12 musicians (string players/keyboardists preferred) + 8–12 musically literate non-experts.
- **Material.** 60–90 s excerpts: (a) baseline closed-form v1, (b) v4/v5 optimized, (c) a high-quality commercial recording excerpt (reference), all level-matched and rendered on the same GM or a neutral sampled set.
- **Design.** Randomized A/B/X with short breaks; headphones or near-field monitors; 44.1/48 kHz, 24-bit.

Rating rubrics (7-point Likert; anchors “much worse” to “much better than reference”):

- **Phrasing naturalness.** Do bar-arches and swells read as musical breathing rather than mechanical pulsing?
- **Texture clarity.** Can you follow three canonic voices distinctly over the ground? (Masking, register separation.)
- **Cadence salience.** How clearly do V→I cadences land (leading-tone lean, resolution satisfaction)?

- **Continuo plausibility.** Does the harpsichord pattern feel stylistically apt (8ths propulsion, 6–5/4–3 colors)?
- **Intonation feel (MIDI illusion).** Despite equal temperament, does the voicing/timing avoid "MIDI glassiness"?
- **Overall musicality.**

Binary probes (Yes/No):

- "Did you notice a suspension resolving at the start of select I bars?"
- "Did period boundaries (every 8 bars) feel like micro-breaths rather than jolts?"
- "Were there moments of obvious octave/fifth 'stacking' that sounded bare?"

Free-text prompts:

- "Name one spot where the phrase shape felt most convincing."
- "Name one spot to thin or thicken texture."

Analysis:

- Within-subject repeated-measures ANOVAs across conditions (v1 vs v4/v5 vs reference) per rubric.
- Report effect sizes and 95% CIs; pre-register exclusion rules (e.g., outliers > 2.5 sd).

6.3 Variants: pure strings, piano reduction, shortened concert cut

Pure strings (no harpsichord).

- **Change.** Remove continuo track; slightly widen dynamic range on strings, add minimal divisi in v3 during dense cycles to retain shimmer without keyboard attack.
- **Objective deltas.** Recompute masking index (e.g., RMS per octave band per voice); expect improved mid-treble separation but less rhythmic bite.
- **Subjective hypothesis.** Higher perceived blend; marginally softer cadence edges; preferred by string-ensemble listeners.

Piano reduction.

- **Change.** Collapse canonic lines to RH (two staves with implied polyphony), ground + implied continuo to LH (broken chords). Add CC64 pedal shapes (half-pedal on cadences; clears on bar 7 and final bars).
- **Objective deltas.** Attack-synchrony tighter; micro-timing amplitude reduced to ~1–2 ticks to avoid flam.
- **Subjective hypothesis.** Clearer harmonic "snapshot" per beat; less contrapuntal independence; good for didactic demos.

Shortened concert cut (10–12 cycles).

- **Change.** Maintain the period-Delta framework; compress R_n schedule to create a single arc: sparse → medium → dense → release. Strengthen final cadence by doubling leading-tone emphasis and widening final ritard (r ~ 0.18).
- **Objective deltas.** Cycle-swell f_cycle(n) becomes asymmetric (faster rise, longer tail).
- **Subjective hypothesis.** Better long-form momentum for 2–3 minute performances; clearer narrative peak.

Comparison matrix (quick checklist):

Variant	Pros	Cons	When to use
Pure strings	Blend, warmth, no key-clicks	Less rhythmic articulation	Chamber strings, live ensemble mockups
Piano	Clarity, portability, pedagogy	Reduced voice independence	Teaching, score study, solo recitals
Concert cut	Strong arc, time-efficient	Less meditative	Short programs, demos, media inserts

Acceptance criteria (ready-to-publish):

- Objective: 0 flagged parallel perfect runs; ≤ 1% non-chord tones on strong beats; registers within [L_k,H_k]; timing/duration constraints satisfied; cadence transforms present at all intended sites.
- Subjective: Mean ratings for v4/v5 ≥ reference−0.5 on "phrasing" and "cadence salience"; ≥ reference on "texture clarity"; ≥ 75% "Yes" on suspension and breath probes.

These checks give us a defensible bridge between the *mathematical sufficiency* of the closed-form and the *musical adequacy* expected by performers and listeners.

7. Generalization

7.1 Other ground-bass canons and passacaglias

Our stack factors into: harmony layer (periodic ground), generator layer (R_n, q_n), delay scheme (Δ_k), idiom table (τ), registration N_k, and expression (ϕ, v). To re-instantiate other ostinato forms, swap only the **periodic ground** and, if needed, the **meter/grid** and **delay**:

- **La Folía (D minor, 3/4).**
  Period $T = 16$bars, meter 3/4, choose $L = 12$(eighth-note triplets) or $L = 24$for finer control.
  Ground (one common version, pc numbers in D minor):
  (i,V,i,VII,III,VII,i,V,III,VII,i,V,i,V,i,V).
  Replace C_seq, set Δto 2 or 4 bars depending on desired imitation spacing, keep τ rules (suspensions, 6–5) but remap scale degrees to minor with leading tone on cadences.

- **Romanesca / Bergamasca (major, 4/4).**
  Period $T = 8$or 16bars, grid $L = 16$. These grounds align closely to our Canon template—often just a re-ordering of {I,V,vi,III,IV}. Swap C_seq; reuse all τ with minimal edits.

- **Chaconne/Passacaglia in 3 (Bach/Handel idiom).**
  Period $T \in \{4,8\}$bars, meter 3/2 or 3/4; choose $L = 12$(sixteenths in 3/4) or $L = 24$(3/2). Often the "ground" is a **descending tetrachord** (e.g., i–VII–VI–V over two bars) repeated. Implement the ground in $root_seq[b]$and admit mixture (♭6 in minor) in A_b.
  τ: emphasize 4–3 over i and Phrygian inflection on V→i in minor.

- **Purcell "Dido" ground (A minor, 3/2).**
  Ground: A–G–F–E (repeating). Encode as $root_seq$with A_b= chordal realizations (i, VII, VI, V). Use $L = 24$, $\Delta = 2$bars for a spacious canon; τ: weight 4–3 resolutions over i and leading tone G# in cadences.

- **Mensuration canons (ratio canons).**
  Keep one $q(t)$but assign **voice-specific time scalings**: $\Delta_k = r_k \cdot \Delta$, $t \mapsto r_kt$for each canonic voice (e.g., 2:3). Registration $N_k$and τ remain local; only the sampling of $q(\cdot)$changes. This yields strict time-ratio canons on the same ground.

Recipe (copy-safe):

1. Choose meter and grid (beats_per_bar, L), compute $g = T_bar/L$.
2. Specify $C_seq[b]$and $root_seq[b]$for one period T.
3. Pick Δ(bars) and, if mensural, scalars r_k.
4. Port τ rules: map 4–3 and 6–5 by scale degree; set leading-tone emphasis at cadences in the new mode.
5. Reuse R_n, q_nor import alternates (7.2).
6. Render with N_k, ϕ, v; verify constraints in §6.1.

7.2 Importing alternative contour libraries R_n, q_n

Our "content" is tiny: per cycle n, a **rhythm set** $R_n \subset \{0, \dots, L-1\}$and a **contour word** $q_n: \{0, \dots, L-1\} \to \{0,1,2\}$. There are three complementary ways to populate these while keeping determinism:

- **Hand-crafted families (deterministic).**
  Provide a palette $\mathcal{R} = \{$quarters,eighths,syncopated,16ths, … $\}$and $\mathcal{Q} = \{$up-arp,down-arp,neighbor-figures,passing-dense$\}$. Then schedule a macro-arc (sparse→dense→release) by a fixed index map $n \mapsto (R_n, q_n)$. This is what we used for Pachelbel.
- **Corpus-derived, then frozen.**
  From a small set of style exemplars (e.g., Corelli/Handel passacaglias), extract histograms of onset patterns and triad-index patterns conditional on chord quality. Fit a **finite, deterministic codebook**: pick the top $KR$and top Kqprototypes via k-medoids on Hamming distance. At runtime, you **select** prototypes by a fixed deterministic schedule (or chord-conditioned lookup), not by sampling—preserving reproducibility.
- **Constraint-solved variants.**
  Pose $q_n$as a short sequence over $\{0,1,2\}$subject to hard constraints (e.g., avoid inner-voice roots on strong beats; require at least one neighbor-tone on weak 16ths in dense bars). Solve by DP/ILP to **one** lexicographically minimal solution per bar class, and cache it. Deterministic, auditable, style-conformant.

Interface (copy-safe):

- $R_n$is a bitmask of length L.
- $q_n$is a length-L array over $\{0,1,2,\perp\}$, with ⊥ignored at off-beats.

- Optional chord-conditioning: $q_n(i) = \pi(A_b, i)$via a 3×L table per chord class (major/minor, root position).
  Swap $(R_n, q_n)$and the rest of the pipeline is unchanged.

7.3 From Pachelbel to parametric "style surfaces"

The closed-form stack exposes low-dimensional **style parameters**. Collect them in a vector $\theta \in \mathbb{R}^d$(many are discrete), and view the renderer as

$$\text{Render}(\theta) \;\Rightarrow\; \text{SMF bytes.}$$

Examples (grouped, copy-safe):

- **Texture & density:**
  θ_dens ∈ {$\mathcal{R}$ index sequence}, $L \in \{12,16,24\}$.
- **Canon geometry:**
  Δ ∈ $\mathbb{Z}$bars, mensural ratios $r_k \in \mathbb{Q}^+$, number of imitative voices V.
- **Idiom weights (discrete toggles):**
  τ_{4–3} ∈ {off, light, strong}, τ_{6–5} ∈ {off, on}, leading-tone bias ∈ {off, on}, strong-beat guard ∈ {off, on}.
- **Registration & voicing:**
  windows $[L_k, H_k]$, inertia bias (minimize |m−prev| vs prefer octave center).
- **Expression:**
  ϕamplitudes $A1, A2$, phase set $\{\theta_c\}$, bar-arch shape f_bar(e.g., sinus vs convex), cycle swell f_cycle, articulation shorten (s_dense, s_sparse), legato overlap ol.
- **Continuo grammar:**
  pattern vector for 8ths, optional hemiola toggle in 3-meter, figured substitutions map.

A **style surface** is then a region $\mathcal{S} \subset \Theta$with constraints (e.g., keep τ_{4–3} active on I bars; bound $|\ \phi\ | \leq 5$ticks; enforce guard on inner voices). Traversing $\mathcal{S}$along a path $\theta(\lambda)$(0→1) yields a **continuous family** of renderings:

- **Baroque→Classical morph:** gradually reduce τ_{6–5}, shrink ϕ, simplify $R_n$toward quarters/eighths, constrain dynamics to narrower arches.
- **Chamber→Virtuoso morph:** expand $L$and densify R_n, widen $[L_k, H_k]$, increase articulation contrast, strengthen cadential leading-tone bias.

- **Pachelbel→Passacaglia morph:** replace C_seq, switch meter $3/4$, adjust Δ, keep idioms; the path is piece-identity-preserving if $C_seq$stays within a curated family of grounds.

Two practical payoffs:

1. **Design space you can *search*** deterministically. Because outputs are bit-stable, you can grid-search $\theta$and evaluate with the objective metrics in §6.1 (e.g., zero parallel-perfect flags, target register occupancy) plus subjective proxies (cadence energy via velocity/tempo alignment).
2. **Pedagogical "explainable sliders."** Each axis has a musical name. Moving a slider changes a *small, local* rule or envelope—not a black-box re-generation—which keeps the analysis reproducible and the edits scrutable.

In short, the Pachelbel encoding is not a one-off—it is a minimal **coordinate chart** on a broader manifold of ground-bass canons and passacaglias. Swap C_seq, import (R_n, q_n), choose Δ(or r_k), and navigate θon a style surface to render families of works with the same deterministic engine.

8. Limitations and Future Work

8.1 Ornamentation density vs. clarity

Our closed-form stack deliberately caps idiomatic density (passing 16ths, suspensions, cadential graces) to keep contrapuntal lines intelligible. This exposes a core tradeoff:

- If we increase weak-beat non-chord tones, introduce mordents/turns, or thicken the continuo with inner neighbors, clarity drops through masking and rhythmic smear, especially when L is high and shorten is small.
- If we thin ornaments and lengthen note values, the texture reads cleanly but risks sounding mechanistic or under-articulated relative to historical practice.

Future work:

1. Density controllers with perceptual feedback. Define a monotone density parameter d in [0,1] that scales the rate of NCT insertion, articulation shorten, and micro-jitter amplitude. Optimize d against objective clarity proxies (e.g., spectral overlap, instantaneous polyphony, voice-crossing count) subject to constraints in 6.1.
2. Local sparsification. Use bar- and beat-level salience maps so cadential spans remain ornament-light while mid-period spans admit more figuration.

3. Register-aware decoration. Permit more ornament only where octave bands are under-occupied, reducing mask risk.

8.2 Figured-bass grammar learning from corpora

Our continuo layer currently uses hand-coded transforms (6-5 over V, 4-3 over I) and a simple 8th-note arpeggiation pattern. This is musically serviceable but under-generalizes.

Future work:

1. Corpus mining to a finite, deterministic codebook. From annotated Baroque corpora (Corelli, Handel, Pachelbel, Purcell), extract local realizations conditioned on chord class, inversion, bass motion, and meter cell. Cluster by discrete features to obtain a small set of realization templates per context. Freeze the top K per context as a codebook and select deterministically via a context hash, preserving reproducibility.
2. Constraint induction. Learn hard constraints (e.g., no doubled leading tone on strong beats in inner voices) and soft preferences (e.g., favor stepwise upper voices during cadences). Encode as guard rules and weighted tie-breakers in N and tau rather than probabilistic sampling.
3. Voice-leading verifiers. Train lightweight classifiers to flag parallel perfects, hidden perfects, and accented dissonances out of place. Use them in-the-loop as rejectors during realization selection, still producing a single deterministic output.
4. Style adaptation without randomness. Allow user-specified style vectors (e.g., Corelli-like vs Purcell-like continuo) that deterministically switch among codebook templates and articulation tables.

8.3 Orchestration beyond strings/continuo

The current render targets a small ensemble (three canonic violins or strings, cello, harpsichord). Broader orchestration raises three issues:

- Timbre mapping. Different instruments project attacks and sustain differently; an oboe or flute canon demands altered articulation shorten, velocity curves, and micro-jitter patterns to avoid machine-like sameness.
- Register stratification. Brass and winds occupy overlapping bands; without care, cadence edges blur or become strident through doubled partials.
- Notational affordances. Harp, guitar, or piano reductions need idiomatic voicing (e.g., rolled chords, idiomatic breaks) rather than literal arpeggio clones.

Future work:

1. Instrument profiles. Define per-instrument articulation envelopes (attack, decay, sustain proxies), preferred dynamic ranges, and phi(t) amplitudes/phases. Profiles become parameters to the same closed-form, enabling e.g., flute trio + cello, or oboe + strings mixtures without changing the core logic.
2. Orchestration constraints. Add a light constraint layer that limits simultaneous onsets per octave band and per instrument family. Enforce color-contrast at cadences (e.g., remove middle doublings at V->I to sharpen resolution).
3. Divisi and doubling policies. Deterministically assign unisons to divisi or second desks when vertical density exceeds a threshold. For larger forces, codify doubling tables (e.g., clarinet on inner fifths, horn on sustained thirds) as local transforms that do not alter harmony or canon geometry.
4. Venue-aware mixing. Provide simple target curves (chamber, hall, broadcast) that reshuffle velocity envelopes and shorten values to compensate for expected reverberation, still yielding the same note grid and preserving bitwise determinism in the SMF event structure.

Across all three fronts, the guiding principle remains the same: keep the generative skeleton minimal and auditable, push stylistic richness into small, local, deterministic tables, and validate with the objective checks in 6.1 plus listening protocols in 6.2.

9. Conclusion

We set out to treat Pachelbel's Canon not as a long sequence of notes but as a compact, auditable program: a closed-form map from discrete time to fully specified musical events. By factoring the piece into a small set of deterministic layers—periodic ground (A_b, root_seq), canonic geometry (Delta-shifted voices), a minimal generator (R_n, q_n), local Baroque transforms (tau for 4–3, 6–5, leading-tone emphasis), registration N_k, and structural expression (phi, v)—we recover the Canon's identity with far fewer degrees of freedom than a traditional score or MIDI list, while retaining the phrasing and cadential clarity listeners expect.

This representation buys four things at once. First, **explainability**: every audible choice corresponds to a named function or table, so edits become principled, global, and reversible. Second, **compression**: minutes of music collapse into a handful of small, copy-safe specifications. Third, **reproducibility**: the rendering is bit-stable—no randomness—yet it preserves a humanlike feel through deterministic envelopes and microtiming. Fourth,

portability: the same core emits library-free SMF suitable for any GM synth or high-end sampler, and it generalizes cleanly to other grounds and meters.

Our evaluation scaffolding bridges analysis and listening: objective checks (voice-leading, register bounds, dissonance placement, tempo plans) gate the output before it reaches subjective panels that assess phrasing, texture clarity, and cadence salience. Variants (pure strings, piano reduction, shortened concert cut) demonstrate that meaningful interpretive change can be expressed as small parameter moves on a shared skeleton rather than wholesale re-authoring.

Limitations remain: ornamentation density must be balanced against intelligibility; figured-bass realization still leans on hand-crafted idioms; orchestration beyond strings/continuo needs instrument-aware envelopes and constraints. Each of these is tractable within the same deterministic ethos—via finite codebooks learned from corpora, guard rules verified by lightweight classifiers, and per-instrument profiles that preserve the underlying map.

More broadly, this work sketches a **style surface**: a low-dimensional, deterministic coordinate chart for ground-bass canons and passacaglias. Swapping C_seq, importing alternative (R_n, q_n), adjusting Delta (or mensural ratios), and dialing expression parameters yields families of musically coherent, bit-reproducible renderings. In that sense, “closed-form Pachelbel” is both a concrete artifact and a template: a practical recipe for turning a historically grounded piece into a small, inspectable program—one that others can read, verify, modify, and extend without sacrificing musicality or rigor.

References

Baroque practice, counterpoint, and figured bass

- Bach, C. P. E. (1753/1949). *Essay on the True Art of Playing Keyboard Instruments* (W. J. Mitchell, Trans.). Oxford University Press.
- Donington, R. (1977). *Baroque Music: Style and Performance*. W. W. Norton.
- Fux, J. J. (1725/1965). *Gradus ad Parnassum* (A. Mann, Trans.). W. W. Norton.
- Gjerdingen, R. O. (2007). *Music in the Galant Style*. Oxford University Press.
- Jeppesen, K. (1939). *Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century* (G. Haydon, Trans.). Dover.
- Niedt, F. E. (1700–1717/1989). *The Musical Guide* (*Musicalische Handleitung*) (P. Poulin, Trans.). Oxford University Press.
- Rameau, J.-P. (1722/1971). *Treatise on Harmony* (P. Gossett, Trans.). Dover.
- Schachter, C., & Aldwell, E. (2011). *Harmony and Voice Leading* (4th ed.). Cengage.

Canons, ostinati, and historical context

- Apel, W. (1969). *Harvard Dictionary of Music* (2nd ed.). Harvard University Press.
- Dreyfus, L. (1987). *Bach and the Patterns of Invention*. Harvard University Press.
- Williams, P. (2007). *J. S. Bach: A Life in Music*. Cambridge University Press. *(general background on chaconnes/passacaglias and ostinato practice)*

Symbolic/algorithmic music and representation

- Cope, D. (1996). *Experiments in Musical Intelligence*. A-R Editions.
- Hiller, L., & Isaacson, L. (1959). *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill.
- Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.
- Taube, H. (2004). *Notes from the Metalevel: An Introduction to Algorithmic Music Composition*. A-R Editions.
- Temperley, D. (2001). *The Cognition of Basic Musical Structures*. MIT Press.
- Temperley, D. (2007). *Music and Probability*. MIT Press.

- Pachet, F., & Roy, P. (2011). Markov constraints for generating musical sequences. *Constraints*, 16(2), 148–169.
- Rohrmeier, M., & Graepel, T. (2012). Comparing feature sets for learning harmonic relations in Western tonal music. *Journal of New Music Research*, 41(4), 329–339.

MIDI, SMF, and performance control

- MIDI Manufacturers Association (MMA). (1996/2014). *MIDI 1.0 Detailed Specification*. MMA.
- MIDI Manufacturers Association (MMA). (1990/1996). *Standard MIDI Files (SMF) Specification 1.0/1.1*. MMA.
- Roads, C. (2019). *The Computer Music Tutorial* (2nd ed.). MIT Press. *(SMF, timing, scheduling overviews)*

Expressive timing, microtiming, and performance modeling

- Gabrielsson, A. (1999). The performance of music. In D. Deutsch (Ed.), *The Psychology of Music* (2nd ed., pp. 501–602). Academic Press.
- Palmer, C. (1997). Music performance. *Annual Review of Psychology*, 48, 115–138.
- Repp, B. H. (1998). Variations on a theme by Chopin: Relations between perception and production of timing in music. *Journal of Experimental Psychology: Human Perception and Performance*, 24(3), 791–811.
- Friberg, A., & Sundberg, J. (1999). Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. *Journal of the Acoustical Society of America*, 105(3), 1469–1484.
- Clarke, E. F. (1988). Generative principles in music performance. In J. A. Sloboda (Ed.), *Generative Processes in Music* (pp. 1–26). Oxford University Press.
- Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press.

Figured-bass realization and continuo practice (focused)

- Buelow, G. J. (2004). *A History of Baroque Music*. Indiana University Press.
- Dirst, M. (2017). *The Continuo Companion*. Oxford University Press.
- Sanguinetti, G. (2012). *The Art of Partimento: History, Theory, and Practice*. Oxford University Press.

Evaluation and listening methodology (useful templates)

- ITU-R. (2015). *BS.1534-3: Method for the subjective assessment of intermediate quality level of audio systems (MUSHRA)*. International Telecommunication Union.
- IEC. (2013). *IEC 60268-13: Sound system equipment—Listening tests*. International Electrotechnical Commission.

Optional primary-score sources (edition-agnostic placeholders)

- Pachelbel, J. (c. 1680). *Canon and Gigue in D major* (various scholarly editions).
- Urtext or practical editions from Bärenreiter, Henle, or Breitkopf & Härtel (for cross-checking voicing and cadential norms).

Note: Citations above are chosen to support (i) Baroque harmony/continuo norms, (ii) canon/ostinato practice, (iii) symbolic/algorithmic representation and SMF engineering, and (iv) expressive timing research. They provide sufficient scholarly grounding for the closed-form modeling, the deterministic MIDI rendering pipeline, and the evaluation protocols described in the paper.

Appendix A — Minimal closed-form spec (all functions and constants)

Pitch-class map (copy-safe): C=0, C#=1, D=2, D#=3, E=4, F=5, F#=6, G=7, G#=8, A=9, A#=10, B=11.

Meter & grid

- beats_per_bar = 4
- Q = ticks_per_beat (e.g., Q=480)
- T_bar = 4*Q (ticks per bar)
- L = 16 (grid points per bar; 16ths)
- g = T_bar / L (ticks per grid)
- T = 8*T_bar (one period = 8 bars)
- Delta = 2*T_bar (canonical delay = 2 bars)

Harmony (D major)

- Ordered triads A_b by bar b in {0..7}:
    - I(D): [2,6,9]
    - V(A): [9,1,4]
    - vi(Bm): [11,2,6]
    - iii(F#): [6,9,1]
    - IV(G): [7,11,2]
- Ground sequence C_seq[b] = [I, V, vi, iii, IV, I, IV, V].
- Bass roots root_seq[b] = [2,9,11,6,7,2,7,9].
- Chord projector P_b(x) = 1 if x in A_b else 0.

Indices

- b(t) = floor(t / T_bar) mod 8
- i(t) = floor((t mod T_bar)/g) in {0..15}
- n(t) = floor(t / T) (cycle index)

Generator per cycle n

- Rhythm set R_n subset {0..15} (e.g., quarters {0,4,8,12}, eighths {0,2,4,6,8,10,12,14}, syncopated, 16ths, etc.).
- Contour word q_n: {0..15} -> {0,1,2} (index into ordered triad).
- Proposed pitch class on onset t_on:
    - pc_prop(t_on) = A_{b(t_on)}[q_{n(t_on)}(i(t_on))].

Local idioms (τ) and guards (applied before registration)

- Suspensions on I bars (b in {0,5}): delay first new attack by 1/8; resolve G(7)->F#(6) on entry.
- Leading-tone emphasis on b=7 (V): force C#(1) at first onset; resolves to D(2) on next bar.
- Continuo 6–5 over V→I: in latter half of V bar, E(4) -> D(2) on selected 8ths.
- Optional continuo 4–3 color over I: early 8ths F#(6) -> E(4).
- Strong-beat guard (inner voices k in {2,3}): on i in {0,4,8,12}, if pc == root(A_b), rotate to next chord member (root→third→fifth→root).

Registration (per voice k)

- Windows [L_k, H_k] (MIDI note numbers), e.g.:
    - k=1 (Vln1): [62,91] (D4..G6)
    - k=2 (Vln2): [59,86]
    - k=3 (Vln3): [55,83]
    - bass: [33,52]
    - continuo: [48,76]
- Register operator N_k(pc, prev) = nearest MIDI in [L_k,H_k] with m % 12 = pc, minimizing |m-prev| (tie-break upward).

Canonic voices

- m_1(t_on) = N_1(pc_adj(t_on), prev_1)
- m_2(t_on) = N_2((m_1(t_on-Delta) mod 12), prev_2)
- m_3(t_on) = N_3((m_1(t_on-2*Delta) mod 12), prev_3)

Bass & continuo

- Bass quarters on root_seq[b(t)], optional passing 8ths on dense cycles (back-8th steps diatonically toward next root).
- Continuo 8ths: [root, third, fifth, third, root, third, fifth, third] with 6–5/4–3 substitutions as above.

Expression

- Micro-timing per channel c:
    - phi_c(t) = round(A1*sin(2*pi*t/(7g) + theta_c) + A2*sin(2*pi*t/(11g) + theta_c/2))
    - small amplitudes (e.g., A1≈1..2, A2≈0.6..1.2 ticks), fixed phases theta_c.
- Velocity envelope:
    - v(t_on) = clamp(V0 * f_bar((t_on mod T_bar)/T_bar) * f_cycle(n(t_on)))
    - f_bar(x) = 0.7 + 0.3*sin(pi*x); f_cycle(n) = 0.9 + 0.2*sin(pi*n/(N_c-1)).

Articulation

- shorten = s_dense if |R_n| large else s_sparse (e.g., s_dense=40, s_sparse=20 ticks).
- min_len = 30..40 ticks; overlap ol = 3..8 ticks for legato.
- Offsets: t_off' = max(t_on' + min_len, t_next - shorten) + epsilon_overlap.

Tempo plan (meta)

- Base bpm (e.g., 92).
- Breath at each period end: set tempo to 0.96*bpm during bar 7, resume at bar 0.
- Final ritard last 2 bars: K stepwise tempo decreases totalling ~12%.

Appendix B — Pseudocode for SMF writer and event scheduling

```
# ====== Precompute ======

constants: Q, T_bar, L, g, T, Delta, bpm

tables:    A_b for b in 0..7, root_seq[b], R_n, q_n, idiom slots

profiles:  [L_k, H_k] per voice, phi_c params, V0, f_bar, f_cycle

tracks:    meta, bass(ch0), cont(ch4), v1(ch1), v2(ch2), v3(ch3)


# ====== Helper: registration ======

function N_k(pc, prev, Lk, Hk):

   candidates = { m in [Lk..Hk] : m % 12 = pc }

   if prev is None: pick m closest to octave center; else pick m minimizing |m-prev|; tie-break higher

   return m


# ====== Idioms (τ) ======

function apply_idioms(pc, role, b, i, Rn, prev_pc, next_chord_class):

   # leading tone on V (bar 7)

   if b == 7 and i == min(Rn) and role in {v1, v2?}: pc := 1       # C#

   # I-bar suspension on entry for v1

   if role == v1 and b in {0,5}:

      if i == 0: mark_hold_previous_to(i=2); return pc

      if i == 2 and prev_pc == 7: pc := 6                    # G->F#

   # continuo 6–5 on V->I

   if role == continuo and chord(b) == V and chord(b+1) == I:

      if i in {8,12}: pc := 4   # E

      if i in {10,14}: pc := 2  # D
```

```
    # optional continuo 4–3 color on I
    if role == continuo and chord(b) == I:
        if i == 2: pc := 6; if i == 6: pc := 4
    return pc

# strong-beat guard for inner voices
function guard_inner_root(pc, role, b, i):
    if role in {v2, v3} and i in {0,4,8,12} and pc == A_b[0]:
        return rotate(A_b, pc)  # root->third->fifth
    return pc

# ====== Voice event synthesis ======
function build_voice_events(role, channel, cycles, total_bars):
    events = []
    prev_pc = None
    prev_m  = None
    for bar in 0..total_bars-1:
        b = bar % 8
        Rn = R_schedule[ floor(bar/8) ]     # choose rhythm set
        qn = Q_schedule[ floor(bar/8) ]     # choose contour word (triad indices)
        onsets = [ bar*T_bar + i*g for i in 0..15 if i in Rn ]
        for idx, t_on in enumerate(onsets):
            i = (t_on % T_bar) / g
            pc = A_b[ qn[idx % len(qn)] ]   # propose chord member
            pc = apply_idioms(pc, role, b, i, Rn, prev_pc, chord(b+1))
            pc = guard_inner_root(pc, role, b, i)
```

```
        # canonic delay for v2/v3 (apply before registration)
        if role == v2: base_pc = (pc_of_v1_at(t_on - Delta)) % 12; pc = base_pc
        if role == v3: base_pc = (pc_of_v1_at(t_on - 2*Delta)) % 12; pc = base_pc

        m = N_k(pc, prev_m, L_role, H_role)

        # timing & articulation
        phi = phi_channel(role, t_on)          # small integer ticks
        t_on2  = t_on + phi
        t_next = next_grid_onset_within_bar(t_on) or (bar+1)*T_bar
        shorten = s_dense if |Rn| >= 8 else s_sparse
        t_off2 = max(t_on2 + min_len, t_next - shorten) + overlap(role)

        # dynamics
        vel = clamp( V0_role * f_bar( (t_on % T_bar)/T_bar ) * f_cycle( floor(bar/8) ), 1, 127 )

        events.append( (t_on2, note_on(channel, m, vel)) )
        events.append( (t_off2, note_off(channel, m, 0)) )

        prev_pc = pc
        prev_m  = m
    return sort_and_pack(events)

# ====== Bass (quarters or passing 8ths on dense cycles) ======
function build_bass_events():
```

```
  events = []
  for bar in 0..total_bars-1:
    b = bar % 8
    dense = is_dense_cycle( floor(bar/8) )
    if not dense:
      for q in 0..3:
        t_on = bar*T_bar + q*(T_bar/4)
        m = N_bass(root_seq[b], prev_m, L_bass, H_bass)
        schedule_note(events, ch0, m, t_on, t_on + T_bar/4, phi_bass, vel=60)
    else:
      for q in 0..3:
        t1 = bar*T_bar + q*(T_bar/4)
        t2 = t1 + T_bar/8
        m1 = N_bass(root_seq[b], prev_m, L_bass, H_bass)
        m2 = N_bass(step_toward(root_seq[b+1], root_seq[b]), m1, L_bass, H_bass)
        schedule_note(events, ch0, m1, t1, t2, phi_bass, vel=62)
        schedule_note(events, ch0, m2, t2, t1 + T_bar/4, phi_bass, vel=56)
  return sort_and_pack(events)

# ====== Continuo (8ths) ======
function build_continuo_events():
  events = []
  for bar in 0..total_bars-1:
    b = bar % 8
    pat = [root, third, fifth, third, root, third, fifth, third]
    for k in 0..7:
```

```
        i   = 2*k
        t_on = bar*T_bar + i*g
        pc = A_b[ pat[k] ]
        pc = apply_idioms(pc, continuo, b, i, Rn_dummy, prev_pc, chord(b+1))
        m  = N_cont(pc, prev_m, L_cont, H_cont)
        schedule_note(events, ch4, m, t_on, t_on + 2*g - 20, phi_cont, vel=56)
    return sort_and_pack(events)

# ====== Meta track ======
function build_meta_track():
    ev = []
    ev.append( (0, tempo_meta(bpm)) )
    ev.append( (0, time_sig_meta(4,2,24,8)) )
    # breaths at end of each cycle
    for cyc in 0..num_cycles-1:
        ev.append( ( (cyc*8 + 7)*T_bar, tempo_meta(0.96*bpm) ) )
        if (cyc+1)*8*T_bar < total_ticks:
            ev.append( ( (cyc+1)*8*T_bar, tempo_meta(bpm) ) )
    # final ritard in last 2 bars (K steps)
    ev += ritard_steps(total_bars-2, total_bars, bpm, r=0.12, K=6)
    return pack_meta(ev)

# ====== Pack SMF ======
function write_smf(tracks):
    header = MThd(format=1, ntrks=len(tracks), division=Q)
    each track = "MTrk" + varlen_len + bytes(events + end_of_track)
```

```
  return header + concat(tracks)
```

Notes:

- sort_and_pack sorts by absolute time, then puts note_off before note_on at the same tick; then converts to delta-time with variable-length encoding.
- schedule_note applies phi, articulation, velocity envelope, and appends on/off pairs.
- All idioms are local; turning any off is a table toggle, not a structural re-render.

Appendix C — Tables for idioms (suspensions, cadences, continuo patterns)

C.1 Suspension & cadence table (upper voices)

Context	Bar b	Grid i (0..15)	Role	Rule	pc’
I-bar entry suspension	0,5	0	v1	suppress attack; hold prev to i=2	—
I-bar resolution	0,5	2	v1	if prev pc==7 then resolve to 6	6
Leading-tone emphasis (V→I)	7	min(R_n)	v1,(v2 opt)	force leading tone	1

C.2 Inner-voice strong-beat guard

Context	Bars	Grid i set	Role	Condition	Action
Avoid root doubling any	{0,4,8,12}	v2, v3	pc == root(A_b)	rotate root→third→fifth	

C.3 Continuo figures (8ths baseline)

Baseline per bar (grid i = 0,2,4,6,8,10,12,14):
[root, third, fifth, third, root, third, fifth, third].

C.3.1 6–5 over V→I

If chord(b)==V and chord(b+1)==I	Grid i	pc’
Suspension “6” (E over A)	8, 12	4
Resolution “5” (D)	10, 14	2

C.3.2 Optional 4–3 color over I

If chord(b)==I	Grid i	pc’
Color tone (4th over bass)	2	6
Resolve to 3rd	6	4

C.4 Bass passing-tone rule (dense cycles only)

Per beat q in {0,1,2,3} within bar b:

- On-beat (i=4q): pc’ := root_seq[b].
- Back-8th (i=4q+2): pc’ := step_toward(root_seq[b+1]) (diatonic neighbor of current root directed toward next bar’s root).

C.5 Articulation and expression defaults

Parameter	Sparse	Dense
Shorten (ticks)	20	40
Min length (ticks)	30–40	30–40
Legato overlap (ticks)	3–5	4–8
Micro-timing amplitude A1	1.0	1.2
Micro-timing amplitude A2	0.6	1.0

Bar envelope: f_bar(x) = 0.7 + 0.3*sin(pi*x)
Cycle swell: f_cycle(n) = 0.9 + 0.2*sin(pi*n/(N_c-1))

C.6 Program/CC defaults (GM)

Part	MIDI ch	Program	CC11 (per bar)
Bass (Cello)	0	42	[82, 98, 86] at [start, mid, end]
Continuo (Hpschd)	4	6	[82, 98, 86]
Vln1	1	48	[82, 98, 86]
Vln2	2	48	[82, 98, 86]
Vln3	3	48	[82, 98, 86]

C.7 Tempo shaping

- Base bpm (e.g., 92).
- Breath at each period end: set 0.96*bpm at tick (cyc*8 + 7)*T_bar, resume at (cyc+1)*8*T_bar.

- Final ritard over last 2 bars: K=6 steps to ~0.88*bpm at final bar end.

Implementation note. The entire pipeline is a small set of tables and copy-safe functions. Every audible choice is traceable to: (C_seq, A_b, root_seq), (R_n, q_n), idiom tables above, N_k, phi_c, v, articulation constants, and the tempo plan.